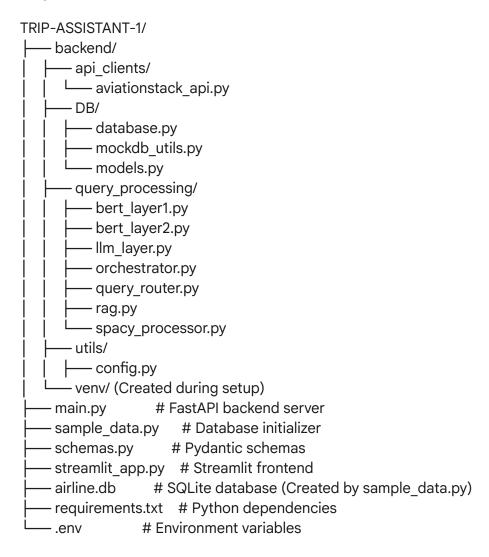
Trip Assistant - Runbook

This document provides instructions on how to set up and run the Trip Assistant chatbot project.

Project Structure:



1. Prerequisites

- **Python:** Ensure you have Python 3.8 or later installed. You can check with python --version.
- pip: Python's package installer. Usually comes with Python. Check with pip --version.
- **Git:** If cloning from a repository.

2. Project Setup

- 1. **Get the Code:** Download or clone the project files into a directory (e.g., TRIP-ASSISTANT-1).
- Navigate to Project Root: Open your terminal or command prompt and change to the project's root directory: cd path/to/TRIP-ASSISTANT-1

3. Create and Activate Virtual Environment

It's highly recommended to use a virtual environment to isolate project dependencies.

1. Create Environment:

python -m venv venv

(This creates a venv folder in your project root).

- 2. Activate Environment:
 - Windows (Command Prompt/PowerShell):
 .\venv\Scripts\activate
 - macOS/Linux (Bash/Zsh):
 source venv/bin/activate

Your terminal prompt should now start with (venv).

4. Install Dependencies

Install all required Python packages using the requirements.txt file.

pip install -r requirements.txt

• Install spaCy Model: The NLP processor needs a language model. python -m spacy download en core web sm

5. Configure API Keys (Optional but Recommended)

Some features rely on external APIs. Configure them using environment variables.

- 1. Create .env file: In the project root directory (TRIP-ASSISTANT-1), create a file named env
- Add API Keys: Open .env and add your keys (replace your_..._key with actual keys):
 # .env
 AVIATIONSTACK API KEY=your aviationstack api key

- AVIATIONSTACK_API_KEY: Get from <u>AviationStack</u> (has a free tier). Needed for live flight status and route search. The code has a default example key, but it might not work long-term.
- OPENAI_API_KEY: Get from <u>OpenAI</u>. Needed for conversational responses (LLM layer) and RAG answers. If omitted, the chatbot will use simpler template-based responses.

6. Initialize Database

Run the sample_data.py script **once** to create the airline.db file and populate it with initial data (customers, flights, policies, etc.). Make sure your virtual environment is active.

python sample data.py

You should see output indicating tables were created and data was loaded.

- Make sure sample data.py includes the desired mock policy text.
- You might want to delete the old airline.db file before running sample_data.py to ensure a clean start with the mock policies.

7. Run the Backend (FastAPI Server)

Start the FastAPI backend server using Uvicorn. Make sure your virtual environment is active.

uvicorn main:app --reload --port 8000

- main:app: Tells Uvicorn to look for the app instance in the main.py file.
- --reload: Automatically restarts the server when code changes are detected (useful for development).
- --port 8000: Runs the server on port 8000 (matching the Streamlit app's default).

You should see output indicating the server is running, usually on http://127.0.0.1:8000. You can visit http://127.0.0.1:8000/docs in your browser to see the API documentation.

8. Run the Frontend (Streamlit App)

Open a **new terminal window** (keep the backend server running in the first one).

- Navigate to Project Root: cd path/to/TRIP-ASSISTANT-1
- 2. **Activate Virtual Environment:** (Activate it again in this *new* terminal)

Windows: .\venv\Scripts\activate

o macOS/Linux: source venv/bin/activate

3. Run Streamlit:

streamlit run streamlit_app.py

9. Access the Chatbot

Streamlit will usually open the application automatically in your web browser. If not, the terminal output will provide a URL, typically:

http://localhost:8501

You can now interact with the chatbot through the web interface.

10. Stopping the Application

- 1. Stop Streamlit: Go to the terminal where Streamlit is running and press Ctrl + C.
- 2. **Stop FastAPI Backend:** Go to the terminal where Uvicorn is running and press Ctrl + C.
- 3. **Deactivate Virtual Environment (Optional):** Type deactivate in each terminal where it's active.

Troubleshooting

- ModuleNotFoundError: Ensure your virtual environment is active and you've run pip install -r requirements.txt.
- Backend Connection Error (Streamlit): Make sure the FastAPI backend server (uvicorn) is running before starting Streamlit. Check that the DEFAULT_BACKEND URL in streamlit app.py matches where Uvicorn is running (usually http://127.0.0.1:8000).
- **Database Errors:** Ensure airline.db was created successfully by running sample_data.py. Check file permissions if necessary.
- API Errors: Verify your API keys in the .env file are correct and haven't expired or exceeded limits. Check the FastAPI backend terminal for specific error messages from external APIs.
- Spacy Model Error: Make sure you ran python -m spacy download en core web sm.